

ユースケースの多面的評価による ソフトウェアモジュールの価値計測

森崎修司^{†1} 高口鉄平^{†2} 永瀬美穂^{†3} 大平雅雄^{†4}

利用者にとって価値の大きなソフトウェアモジュールから順番に開発コスト(労力)を優先的に割当てることにより、ソフトウェア開発の費用対効果を大きくすることができる。しかし、まだ開発が完了しておらず利用できないモジュールがもたらす価値を評価するのは容易ではない。本研究では、ソフトウェアの価値をソフトウェアモジュール毎に計測する手法を提案する。提案手法では、まず、ソフトウェアが提供する価値を価値要素として定義し、その価値の計測方法を定義しておく。次に、ソフトウェアのユースケースを列挙し、それぞれのユースケースの価値を定義しておいた方法で評価する。ユースケースとモジュールの対応を調べ、モジュールごとに価値を計測する。価値要素別に整理すること、及び、ユースケースをもとにモジュールごとに価値を計測することにより、どれくらいの開発コストを費やすのが妥当であるかを推測しやすくなることが期待される。

An approach for measuring values of software module by evaluating multidimensional attribute of the use cases

Shuji Morisaki^{†1} Teppei Koguchi^{†2} Miho Nagase^{†3} Masao Ohira^{†4}

Assigning larger development effort to software modules with larger values potentially increases cost effectiveness. However, estimating values of software modules is not easy before the modules are developed and able to be executed. This paper proposes an approach for estimating the values of the modules. First, our approach predefines value vector and its basis for evaluation. The value vector consists of value elements. Then, all use cases of the software are enumerated. Each use case is evaluated to obtain the corresponding value vector. Finally, each value vector of use case is aggregated into value vectors of corresponding software modules.

1. はじめに

ソフトウェアで実現するサービスやシステムがビジネスの軸を支えるようになり、サービスやシステムから得られる対価と開発にかかったコストを評価し、費用対効果を明らかにすることが求められることも増えてきた。また、ソフトウェアが提供する機能のうち、実際に利用されている機能が10%程度であるという報告もある[5]。しかし、開発コストをどの機能に割当てるかを定める優先順位付けは容易ではない。そのため、機能に優先順位をつけるために analytic hierarchy process[6]を用いた研究[3]をはじめとして様々な支援手法が提案されている。文献[3]の手法では、開発関係者が任意の二つの機能を比較してどちらを先に実装すべきかを回答すると、実装する優先順位が大きな機能から順番に並べた結果を得ることができる。

機能ではなく要求を優先順位付けすることによって開発コストの割当ての優先順位をつけようとする手法や研究も多くある。たとえば、ユースケースの優先順位付け方法の比較[4]、ユーザストーリー[1]の優先順位付け[2]といった

研究である。しかし、これらの研究での優先順位付けや価値の評価は信頼性や利用性のようなソフトウェア品質特性ごとの評価であったり利用者の直感的な要望であったりした。そのため、評価の対象はソフトウェア(アルゴリズムやプログラム)が中心であり、同一ソフトウェアの利用者が多いことによるネットワーク効果やサービスやシステムが提供するコンテンツの豊富さといった点での評価には言及していない。

本研究では、アルゴリズムやプログラムが提供する価値に加え、サービスやシステムが持つネットワーク効果やコンテンツの豊富さを価値に反映するための価値計測の方法を提案し、限られた開発コストを価値の高い機能の開発に割当てられるようにする。具体的には、あらかじめ定義しておいた価値要素と価値の計測方法により、サービスやシステムの個々のユースケースを価値要素ごとに評価する。ユースケースを実現するモジュールやサブシステムごとに価値を合算し、もっとも価値の大きなモジュールやサブシステムから順に開発コストを割当てる。

以降2章で提案手法の実施手順を示す。3章では提案手法の主要な手順である価値計測の例を示す。4章で提案手法を仮想的に動画サービスに適用した例を示し、5章で考察し、6章でまとめる。

†1 名古屋大学

Nagoya University

†2 静岡大学

Shizuoka University

†3 産業技術大学院大学

Advanced Institute of Industrial Technology

†4 和歌山大学

Wakayama University

2. 提案手法

2.1 概要

提案手法による価値評価は大きく 4 つの手順から成る。
 (手順 1) 評価観点である価値要素 $v = (v_1, \dots, v_i, \dots, v_m)$ と v_i の評価方法を決める。

(手順 2) 評価対象のシステムのユースケース $U = \{U_1, U_2, \dots, U_n\}$, システムのモジュール(構成要素) $M = \{M_1, M_2, \dots, M_j\}$, ユースケース $U_i (\in U)$ を実現しているモジュールの集合 $M_{U_i} = \{M_p, M_q, \dots\}$ を得る。

(手順 3) ユースケース $U_i (\in U)$ に対して価値要素 v を評価する。

(手順 4) M_{U_i} と U_i の v から $M_j (\in E)$ の価値 v を得る。

本章の以降では、手順に沿って詳細を述べる。

2.2 価値要素の決定(手順 1)

複数の観点から価値を評価する。 v は評価対象となるソフトウェアやシステムの利用者によって求められる価値に応じて設定するベクトルである。ベクトルの個々の要素が価値要素に対応する。価値要素ごとに価値の評価方法も同時に定義する。本研究では価値要素やその評価方法に前提はおかないが、手順 4 でのシステムのモジュール毎の集計が可能であるものを前提とする。なお、 v の要素は必ずしも独立である必要はないが、類似の要素 v_i, v_j がある場合には、それぞれの価値をどのように評価するかを決めておく。

本節の以降では、3 つの基本価値要素と対応する価値要素の評価方法のタイプを述べる。

2.2.1 基本価値要素

基本価値要素は表 1 に示す v_1, v_2, v_3 である。 v_1 はアルゴリズムによる価値である。アルゴリズムやプログラムによって利用者が得る価値である。たとえば、手作業で集計しなければならなかった部分を自動化していたりカメラで撮影した画像に映っている人物を認識したりといった機能が該当する。

v_2 はソフトウェア利用者のネットワーク効果による価値である。ネットワーク効果は、ソフトウェアの利便性が利用者の数によって影響を受けることを指す。たとえば、ある特定のワードプロセッサアプリケーションソフトウェアのファイル形式は、利用者が増えるほど文書の共有がしやすくなる。その結果、そのワードプロセッサの利便性が大きくなり、そのワードプロセッサを利用する利用者のインセンティブが大きくなる。また、ファイル共有サービスのようなサービスでもアカウントを取得している利用者が増えると利便性が大きくなる。ファイル共有に先だってアカウントを作成する必要や使い方を学習する必要がなくなるからである。

v_3 はソフトウェアが提供するコンテンツによる価値である。たとえば、会計システムやブログサービスでは、帳簿テンプレートやブログ背景画像の質や量が、その利便性に

表 1 基本価値要素

Table 1 Value vector elements

	価値要素の名称	説明
v_1	アルゴリズムによる価値	アルゴリズムやプログラムによって実現される利用者の利便性
v_2	ネットワーク効果による価値	システム利用者の規模がもたらす利便性
v_3	コンテンツによる価値	コンテンツ自体がもたらす利便性

表 2 価値要素の評価方法

Table 2 Evaluation type for value vector

	顕示選好法	表明選好法
長所	価値の評価(推計)に市場データ等を使える。	未利用価値を評価(推計)できる。
短所	未利用価値など顕示されない価値を評価するのが難しい。	調査、分析方法を誤ると信頼性を欠く可能性がある。

影響を与える。

v_1, v_2, v_3 以外にも、所有、利用していること自体に価値があるといった価値要素を定義し、価値ベクトル v を追加することができる。

2.2.2 価値の評価方法

それぞれの価値要素に対応する v_i はソフトウェアのユースケースから求める。本研究では、価値ベクトル v を求められる方法であれば、価値の評価方法に特定の方法を前提としない。開発関係者で集まって、ユースケースごとに相談し類推して決めてもよい。また、既存の調査手法を用いることもできる。

既存の手法を用いる場合、ユースケースからの価値要素の評価方法は大別して顕示選好法と表明選好法の二つがある[7][8]。情報や調査に費やせるコストに応じて選ぶ。二つの方法の概要は次のとおりである。また、それぞれの長所と短所を表 2 に示す。

顕示選好法は、利用者の過去の利用実績や行動から評価する。たとえば、あるサービスで提供される択一のユースケース U_α, U_β に対して、同一の価格がついていれば、利用者がどちらを選んだかによって、ユースケースの価値を推計できる。見積もりたい U_γ に似たユースケースを U_α や U_β といった利用者が過去に選択した実績のあるユースケースから選び、 U_γ の価値を推計する。顕示選好法は入手可能な実績データを使える点が利点である。一方でまだ実現されていない価値を予測するためには慎重な議論が必要である。

表明選好法は、まだ実現されていないユースケース U_γ が提供されたときにどのくらいの対価を支払うつもりがあるかを利用者アンケート等で聞くことにより、価値を評価する。表明選好法は、まだ実現されていない価値を評価

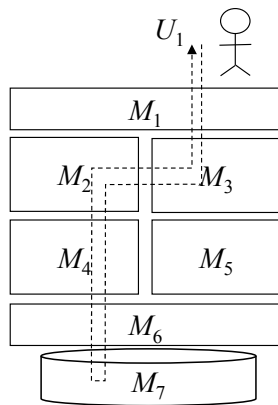


図1 ユースケースを実現するモジュールの対応の例

できる利点がある一方で、調査や分析方法を誤ると信頼できない価値評価となる場合がある。

2.3 ユースケースとモジュールの対応づけ(手順2)

ソフトウェアのユースケース U はシステム開発において要求を整理したときのものを使う。本研究ではユースケースの記述方法に特に前提は置かないが、手順3においてユースケースの価値が評価できるものであることを想定している。たとえば、ソフトウェア開発中に作成したユーザストーリーやユースケースシナリオが該当する。本論文の以降では、利用者の視点から見たシステムの利用手順を自然言語で記述したものをユースケースとする。

ユースケース U_i を実現しているモジュールの集合 M_{U_i} はユースケースを利用した際の実行フローから得る。図1はモジュール M とユースケース U の関係の例を示したものである。点線はユースケース U_1 の実行時に必要となるモジュールをたどったものである。図1はユースケース U_1 の実行にはシステムのモジュール $M_1, M_3, M_2, M_4, M_6, M_7$ が必要であることを表している。つまり $M_{U_1} = \{M_1, M_2, M_3, M_4, M_6, M_7\}$ である。

2.4 ユースケース評価(手順3)とモジュール評価(手順4)

ユースケース U_i を価値 v のそれぞれの要素と評価方法で評価する。評価方法は手順1で定義したものを使う。

ユースケース U_i の v と実現しているモジュールの集合 M_{U_i} から M_j の価値 v を求める。 M_j には、価値要素ごとの価値が評価される。たとえば、検索機能を実現するモジュール M_j に対して、基本価値要素のアルゴリズム v_1 とコンテンツによる価値 v_3 において大きな価値があると評価されたとする。すると、検索機能には、適切なアルゴリズムの実現が必要であり、実現に開発コストを割当てることが妥当であると判断できる。さらに、コンテンツによる価値 v_3 に大きな価値があるとわかれば、検索対象のコンテンツが豊富にあることはもちろんのこと、コンテンツが大量にあった場合でも、検索アルゴリズムが適切に実現できることも

意味している。すなわち、 v_1 と v_3 の価値が大きいことが明らかになることにより、 M_j の開発においてどういう点に注力すればよいかも明らかになる。

3. 価値の評価方法の例

2.2.2節で述べたとおり、提案手法では価値の評価方法として特定の手法を前提としないが、本章では既存の調査方法を用いた評価方法として、顕示選好法と表明選好法の例を述べる。

3.1 顕示選好法

顕示選好法での価値評価方法として次のような回帰式を用いた分析が考えられる。

$$\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} + \begin{pmatrix} \beta_{11} & \cdots & \beta_{1v} \\ \beta_{21} & & \vdots \\ \vdots & \ddots & \vdots \\ \beta_{m1} & & \beta_{mv} \end{pmatrix} (X_1 \cdots X_v) + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_m \end{pmatrix}$$

ここで、 v は価値ベクトル(手順1で定義した価値要素をもつ)、 α は定数ベクトル、 X は実績データとしての利用履歴や購入履歴などの価値要素を構成する各要因、行列 β は実績データ X の各要因の係数、 ε は誤差項である。

X の要素は必ずしも量的変数である必要はない。特定機能の購入の有無のような質的変数を用いることもできる。評価は複数ケースのクロスセクションデータ、特定ケースの時系列データの両方を使うことができる。それぞれ最小二乗法や一般化最小二乗法を使い分けることを想定している。

有料会員向けの SNS(ソーシャルネットワークサービス)を例に回帰式を説明する。被説明変数である価値ベクトル v は金額で表す。たとえば、売上高や当該サービスの利益を用いる。最終的に開発コストとの換算が必要になるためである。ただし、金額以外に開発コストとの変換が容易なものであれば、金額である必要はない。

タイムラインに表示する投稿内容に明示的にフィルタをかけたかこれまでの閲覧履歴にもとづいた利用者の興味に合わせて表示したりすることは、 X_i の一要素となる。その際のフィルタや利用者の興味に合った内容を選ぶことは、 v_1 のアルゴリズムの価値に該当する。時系列に表示すると特定の友人の投稿のみで埋め尽くされる可能性もある。こうした要素を質的変数として X の一要素とする。

SNS にアカウントを作成している利用者の数や知人の数はネットワーク効果 v_2 そのものである。利用者数として X の一要素とする。

v_3 に該当するものとして、投稿は利用者が入力したテキストのみ、SNS が用意する絵文字を利用できる、アルバム集や訪問した場所を地図で示せるような投稿テンプレート、動画用のテンプレートやサンプル動画、といったものが該当する。これらについても、質的変数として X の一要素と

する。

このようなデータセットを分析することで、有料 SNS の価値について、どの要素がどの程度影響を与えているかをみることができ、評価しようとするユースケースの価値を得るための手がかりとすることができる。

3.2 表明選好法

表明選好法の例として、コンジョイント分析により WTP(Willingness To Pay)を求める方法がある。コンジョイント分析は、価値を構成する様々な要素を“属性”として設定したうえで、各属性がどのような値をとるかという“水準”を設定する。そして、各属性のさまざまな水準の組合せによるユースケースを複数利用者に提示し、もっとも望ましいものを選択してもらうことを繰り返すことで、各属性の重要度を推計する。属性のひとつとして価格を設定しておくことにより、各属性や、ユースケースシナリオに対する WTP を推計することができる。

コンジョイント分析では、はじめに利用者 i が j の選択肢から k を選択する確率を示す確率変数をつぎの式を用いて得る。ここで X は選ばれた選択肢におけるベクトル、 g はパラメータ β の分布関数、 θ はパラメータである。

$$P_{ik} = \int \frac{\exp(\beta_i' X)}{\sum_j \exp(\beta_j' X)} g(\beta|\theta) d\beta$$

つぎに、たとえば各属性に関する変数に基づくモデルの効用関数をつぎのように特定化し、推計する。ここで、 X は各属性、 β はパラメータ、 ε は誤差項である。

$$U_{ij} = \sum_k \beta_{ki} \times X_k + \varepsilon_{ij}$$

複数の属性、水準が設定されるユースケースにおいて、その組み合わせは膨大になる可能性があるが、コンジョイント分析においては、直交計画法によるプロファイルの絞り込みによって、いくつかの水準の組合せを利用者に選択させることで推計が行える。

コンジョイント分析について、具体的にはつぎのように進められる。前項と同様の有料型 SNS を例に挙げれば、はじめに属性と水準を表 3 のように決定する。このケースでは、 $2 \times 4 \times 4 \times 4$ の 64 通りの組み合わせが考えられるが、直交計画法を用いて 16 のプロファイル(組合せ)に絞り込むことができる。16 プロファイルの中からいくつかを提示し、そのなかで最も望ましい選択肢を回答させるといったことを繰り返しデータとして収集し、例えばランダム・パラメータ・ロジットモデルなどより評価する。評価を行った結果、各属性の係数が重要度となる。また、価格の係数で他の係数を除すことにより、それぞれの属性への WTP を求めることができる。

なお、上記の推計方法からあきらかなように、類似のユースケースの実績から推計する回帰モデルであっても、見込み利用者へのアンケートからするコンジョイント分析で

表 3 SNS での属性と水準の例

Table 3 Examples of a set of attribute and level in an SNS

属性	水準
投稿の表示	全投稿時系列, フィルタ・重みづけあり
加入時利用者数(人)	100, 1000, 1万, 10万
投稿形式	テキストのみ, 絵文字, 投稿テンプレート, 動画
月額利用料 (円)	200, 400, 500, 1000

あっても、条件によってはソフトウェアのユースケース U ごとに、 v_1, v_2, v_3 を同時に推計することが可能である。回帰モデルであれば、各 v 部分を表現する変数を同時にモデルに設定する。コンジョイント分析であれば、各 v の属性を独立して設定する。

4. 利用シナリオ

4.1 概要

本節および次節では、提案手法の具体例として一般的なサービスへの適用を例示する。なお、価値の評価方法は 3 章の顕示選好法や表明選好法ではなく、開発関係者で相談しながら、類推したものとしている。

本章で対象とするサービスは動画再生サービスである。Youtube のように無料でアップロードでき、再生も無料でできる。サービス提供者は再生時に挿入される広告費用を収入源としている。

このサービスを構成するモジュールの価値を提案手法を用いて評価する。サービスを構成するユースケースは、 U_1 動画サービスのリンクをたどって動画を視聴するユースケース、 U_2 動画再生終了後に一覧表示される類似の動画の推薦から次に再生する動画を選ぶユースケース、 U_3 動画にコメントを書き込むユースケース、とする。以降では、これらのユースケースを次のように表記する。

- U_1 : 動画の視聴
- U_2 : 動画の推薦
- U_3 : コメントの書き込み

また、本サービスを構成するモジュールは次の 9 つとする。

- M_1 : 動画再生インタフェース
- M_2 : コメント付与インタフェース
- M_3 : 動画ストリーム受信モジュール
- M_4 : 動画ストリーム送信モジュール
- M_5 : 動画推薦モジュール
- M_6 : 動画コンテンツ
- M_7 : 視聴履歴
- M_8 : コメント

ユースケース U とモジュール M の関係を図 2 に示す。図中の四角、及び、円筒はモジュールを表す。矢印はユースケースでの情報の流れを示している。矢印についたラベ

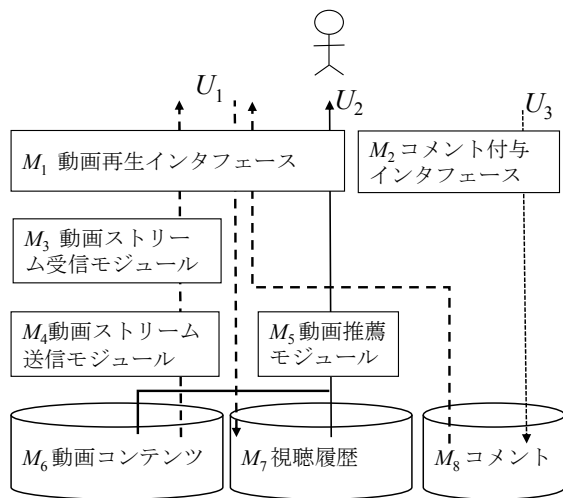


図2 動画サービスのユースケースとモジュールの対応

表4 ユースケースと価値の例

Table 4 Example of use cases and corresponding value

ユースケース	v_1	v_2	v_3
U_1 動画の視聴	630	-1200	160
U_2 動画の推薦	180	50	35
U_3 コメント付与	20	-80	15

ルはユースケース U_1, U_2, U_3 に対応する。たとえば、ユースケース U_3 は、コメント付与インタフェース M_2 、及び、コメント M_8 から実現されることを示している。なお、本利用シナリオではコンテンツをモジュールとして扱う。コンテンツを別扱いにしてもよい。

4.2 価値要素の決定(手順 1)

価値要素は表 1 に示した 3 つの基本価値要素とする。価値の計測方法は、既存の調査手法を使うのではなく、開発関係者間で相談しながら類推するものとする。

4.3 ユースケースとモジュールの対応付け(手順 2)

図 2 に示したとおり、ユースケースを実現するモジュールは次のとおりである。

- $M_{U1} = \{M_1, M_3, M_4, M_6, M_7, M_8\}$
- $M_{U2} = \{M_1, M_5, M_6, M_7\}$
- $M_{U3} = \{M_2, M_8\}$

4.4 ユースケース評価(手順 3)

本利用シナリオでは、3 章で述べた調査方法ではなく、開発関係者がカジュアルに価値を類推する方法でユースケースを評価し WTP を調べたものとする。以下はその過程である。

U_1 について考える。ここでの v_1 は安定して動画が閲覧できることが v_1 に対応する。 v_2 については、動画サービスの利用者規模によってもたらされる価値が該当するが、 U_1 においては v_2 が大きな影響を与える。動画サービスにおける動画の視聴においては、そもそもどの程度の動画が蓄積さ

れているかが価値に影響を大きく与えるからである。厳密には、動画を投稿する利用者の数が増えるにしたがい動画視聴の価値は高まる。また、広告をベースとしたビジネスモデルであるため動画視聴者数が増えるにしたがい動画ファイルをアップロードしている利用者数にとっての価値も高まるという間接ネットワーク効果がある。

v_3 は、動画コンテンツの質と量である。動画コンテンツの質と量は利用者の数が増えると大きくなるのが予想されるので v_2 のネットワーク効果とも相関がある。

さらに U_2, U_3 についても同様に v の分析を行おうとすれば、つぎのようにみることができる。

- U_2 : 動画の推薦
 - v_1 : 推薦する動画を選ぶアルゴリズムによってもたらされる価値
これは、 U_1 と同様、推薦が利用者にとってどの程度有益かによって変動する価値となる。
 - v_2 : 動画推薦の利用者によってもたらされる価値
動画推薦のユースケースでは、他の利用者の視聴履歴を使って動画を推薦するので、 v_2 は大きくなる。ただし、 v_1 よりも与える影響は小さい。
 - v_3 : 動画の充実により高まる推薦の価値
動画コンテンツが充実していると推薦の価値が高まる可能性が高い。 U_1 の v_3 よりも小さい価値となるが価値はある。
- U_3 : コメントの書き込み
 - v_1 : コメント書き込み機能(アルゴリズム)によってもたらされる価値
動画に関連づけてコメントできること、コメントの書きやすさ、表示形式といったユーザビリティが影響する価値である。
 - v_2 : コメント書き込みの利用者数によってもたらされる価値
他者のコメントの存在がどの程度自身のコメントの書き込みのインセンティブとなるかという点を分析することでネットワーク効果の影響を調べる。
 - v_3 : コメントの充実によってもたらされる価値
これも動画の推薦ユースケースと同様、 U_1 の v_3 よりも小さい価値となることが推測されるが、価値はある。

これらの議論から、表 4 に示すユースケース毎の価値が得られたとする。表 4 の値は金額であり、単位は円である。

4.5 モジュール評価(手順 4)

手順 2 と手順 3 で得たユースケースの価値からモジュールの価値を得る。結果を表 5 に示す。なお、実現しているユースケースの価値をモジュールごとに単純に足し算している。

表 5 ユースケースと価値の例

Table 5 Example of use cases and corresponding value

モジュール	ユースケース	v_1	v_2	v_3
M_1	U_1, U_2	810	-1150	195
M_2	U_3	20	-80	15
M_3	U_1	630	-1200	160
M_4	U_1	630	-1200	160
M_5	U_2	180	0	35
M_6	U_1, U_2	810	-1150	195
M_7	U_1, U_2	810	-1150	195
M_8	U_1, U_3	650	-1280	175

5. 考察

提案手法により、ユースケースの価値を多面的に評価し、ユースケースを実現するモジュールごとに積算することで、モジュールの開発に投入できる開発コストを見積もる。価値の類推は3章に示したような統計的な調査手法でも4章に示したような開発関係者の相談により類推するカジュアルな方法でも実現できる。実際のユースケースやモジュールでの検討は重要な今後の課題である。

価値の評価方法において、顕示選好法をA/Bテストの結果やサービスの利用履歴から収集することにより、価値評価の手間を大幅に低減することができる。4章では価値の評価結果を金額としたが、利用者が使った時間や利用者の訪問回数といった指標を用いることもできる。そうした価値評価の一部を自動化する方法も今後の課題である。

4章の表5に示したように、複数のモジュールにおいて同一の評価値が得られたものがある。現実のサービスやシステムではユースケース、モジュール数とも大きくなるものが予測されるため、こうした同一の評価値は減るものと考えている。複数のモジュールで価値評価結果が同一になるモジュールの数がどの程度あるかを調査することも今後の課題である。また、ユースケース、モジュールが増えたときに必要となる手間の低減も今後の課題である。

6. まとめ

利用者にとって価値の大きなソフトウェアモジュールから順番に開発コスト(労力)を優先的に割当てることにより、ソフトウェア開発の費用対効果を大きくすることを目的として、ソフトウェアモジュールの価値の評価手法を提案した。提案手法では、ソフトウェアの価値をソフトウェアモジュール毎に評価し計測する。まず、ソフトウェアが提供する価値を価値要素として定義し、その価値の計測方法を定義しておく。本論文では、基本的な価値要素として、アルゴリズムやプログラムによる価値、利用者のネットワーク効果による価値、コンテンツの豊富さによる価値の3つを提案した。

次に、ソフトウェアのユースケースを列挙し、それぞれのユースケースの価値を定義しておいた方法で評価する。

ユースケースとモジュールの対応を調べ、モジュールごとに価値を計測する。価値の計測には、既に利用実績のある既存の類似ユースケースから推測する顕示選好法と将来提供されるユースケースに利用者がどの程度の対価を支払うかをアンケート等によって調査する表明選好法の2タイプから選ぶ。

最後に、ユースケースごとの価値を合算しモジュールごとの価値を求める。ユースケースを実現しているモジュールの対応はあらかじめ求めておく。モジュールには価値要素ごとに優先順位をつけることができる。たとえば、基本価値要素のアルゴリズムやプログラムによる価値が大きければ、アルゴリズムやプログラムの洗練に開発コストを割当てることでコスト対効果を大きくすることができる。また、ネットワーク効果が大きいモジュールでは、利用者数が増えても対応できるようなしくみをモジュールに持たせるために開発コストを割当てることでコスト対効果が大きくなるのが期待される。このようにして、個々のモジュールに対してどういった側面で開発コストを割当てればよいかを明らかにすることができる。

価値要素別に整理すること、及び、ユースケースをもとにモジュールごとに価値を計測することにより、どれくらい開発コストを費やすのが妥当であるかを推測しやすくなるのが期待される。今後の課題は実際のデータをもとに提案手法を評価することである。

参考文献

- 1) K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley Longman Publishing(1999)
- 2) N.C. Haugen, An Empirical Study of Using Planning Poker for User Story Estimation, Agile Conference 2006, pp.23-28(2006)
- 3) J. Karlsson and K. Ryan, A Cost-Value Approach for Prioritizing Requirements, IEEE Software, vol. 14, no. 5, pp. 67-74(1997)
- 4) J. Karlsson, C. Wohlin, B. Regnell, An Evaluation of Methods for Prioritizing Software Requirements, Journal of Information and Software Technology, vol. 39, no. 14-15, pp. 939-947(1998)
- 5) D. Rice, Geekonomics the real cost of insecure software, Pearson(2007)
- 6) T.L. Saaty, The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation, McGraw-Hill (1980)
- 7) 木下栄蔵, 大野栄二, AHPとコンジョイント分析, 現代数学社(2004)
- 8) 依田高典, ブロードバンド・エコノミクス, 日本経済新聞社(2007)